An Object Tracking in Particle Filtering and Data Association Framework, Using SIFT Features

Malik Souded, Laurent Giulieri, Francois Bremond

An Object Tracking in Particle Filtering and Data Association Framework, Using SIFT Features

Malik SOUDED[1,2], Laurent GIULIERI[1], François BREMOND[2]

[1]Digital Barriers, Sophia Antipolis, France
[2]Pulsar team, INRIA Sophia Antipolis-Méditerranée, France
[1]{Malik.Souded, Laurent.Giulieri}@digitalbarriers.com
[2]{Malik.Souded, Francois.Bremond}@inria.fr

Keywords: SIFT, Particle filtering, Objects tracking, Video surveillance.

Abstract

In this paper, we propose a novel approach for multi-object tracking for video surveillance with a single static camera using particle filtering and data association. The proposed method allows for real-time tracking and deals with the most important challenges: 1) selecting and tracking real objects of interest in noisy environments and 2) managing occlusion. We will consider tracker inputs from classic motion detection (based on background subtraction and clustering). Particle filtering has proven very successful for non-linear and non-Gaussian estimation problems. This article presents SIFT feature tracking in a particle filtering and data association framework. The performance of the proposed algorithm is evaluated on sequences from ETISEO, CAVIAR, PETS2001 and VS-PETS2003 datasets in order to show the improvements relative to the current state-of-the-art.

1 Introduction

Real-time object tracking is an important and challenging task in Computer Vision. Among the application fields that drive development in this area, video-surveillance has a strong need for computationally efficient approaches that combine real-time processing with high performance. Proposed solutions must be able to adapt to different environments and levels of noise and to track with precision a large variety of objects.

In the video surveillance context, many object tracking techniques have been proposed [1, 2, 3, 4, 5]. These techniques can be classified according to three criteria:

The first concerns the initialization of the tracking targets. Many approaches have been tried, from object detection on static images, to initialization by motion detection and clustering, through learned models on whole or parts of objects. In [6], Breitenstein et al. track people using continuous confidence of pedestrian detectors [7, 8] and online trained classifiers. In [9] Moutarde et al. use "connected control-points" features with adaboost for detecting and tracking vehicles and pedestrians. Wu et al. use human body parts detectors in [10]. These parts detectors are trained by boosting a number of weak classifiers based on edgelet features. Siebel et al. [11] use motion detection to detect moving regions, detect and track heads on these regions, and finally track human shapes.

The second criterion concerns the type of features used for the characterization and matching of objects over time. Among all existing features in the state-of-the art, local features are widely used for their accuracy, stability and invariance against scale, rotation and illumination changes within the images and for the affine transformations they can provide. We can mention SIFT[12] and its derivatives PCA-SIFT[13], GLOH[14] and DAISY[15]. Other local features like SURF[16], HOG[7], and BRIEF [17] use similar concepts but they differ on the type of information used (gradient or integral image), the size and shapes (rectangular or circular) of computing regions around points of interest or the normalization and weighting technics.

The last criterion concerns the technique for searching and matching features over time. The most commonly encountered techniques are based on filtering. The oldest and most well-known is the Kalman filter. More recently, many increasingly sophisticated techniques were used - including Particle filters[18, 19, 20, 21, 22]. In [23] Almeida et al. detect and track multiple moving objects using particle filters to estimate the object states, and sample-based joint probabilistic data association filters to perform the assignment between the features and filters. Rui et al. propose in [24] an unscented particle filter to generate sophisticated proposal distributions to improve the tracking performance. Nummiaro et al. [25] integrate an adaptive colour distribution to model targets into particle filtering for object tracking purposes.

Our approach uses moving objects as input. These objects are detected using background subtraction and clustering methods. Once objects are detected, we track them using particle filtering applied to SIFT features and with a specific data association method based on the tracked SIFT features.

This paper presents the following contributions: 1) A novel approach for object tracking in a particle filtering and data association framework. 2) We exploit the high reliability of SIFT features to perform an initial tracking using a particle filter. This is done in a particular way, based on more precise feature detection and selection. 3) In order to deal with less reliable SIFT features and the complex situations that can occurs during object tracking, we propose a novel approach for data association, based on a reliability measure of tracked SIFT features, computed during the particle filtering step. 4) We evaluate the proposed approach on several datasets demonstrating that it is applicable in a video surveillance context and provides interesting results.

2 Our approach

We first present an overview of the algorithm. Our approach consists of two collaborating levels of processing. First, from detected objects at time *t*, denoted *do(t)*, a set of SIFT[12] features is extracted according to criteria detailed in Sec 2.1. All SIFT features are tracked over time using a particle filter and their states are updated at each frame. We will explain what the "state" of the SIFT feature is in Sec 2.2. The next step is the updating of the state of tracked objects of interest at time *t-1*, denoted *to(t-1)*, using the tracked SIFT features. A reasoning based on weighted scoring is introduced to minimize the error due to the effects of SIFT features detected on the background or diverged from the correct object during tracking. In this step, occluded objects are referenced and maintained for tracking resumption. Finally, new detected objects are used to initialize new tracked objects (see Figure 1).

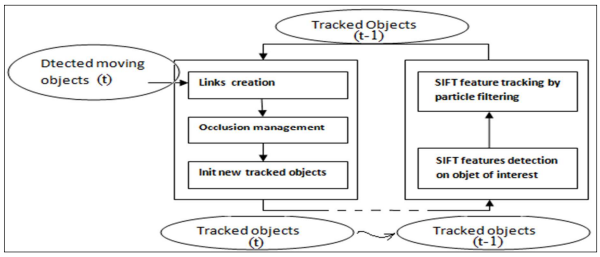

Figure 1: A diagram of our object tracking framework

2.1. SIFT feature detection

For each object in *do(t)*, defined by a bounding box and a set of motion pixels, our system detects and computes a set of SIFT features on this object. The bounding box is divided into small rectangular sub-regions. The aim of this subdivision is to obtain a good spatial distribution of features on the object, allowing for better partial occlusion management (see sec 3.1.1). Another benefit is the possibility of parallelization of computing per sub-region for real time processing using multi-core processors. The number of sub-regions is calculated according to the bounding box dimensions to ensure the robustness and optimize the processing time. Each sub-region must contain a constant number of SIFT features. A SIFT detector with more permissive curvature and contrast thresholds than optimum ones [12] is used to obtain more SIFT points (see sec 3.1.1). The needed number of features is selected according to their robustness based on the detection scale, the curvature and contrast values. This selection is also done using the motion state of pixels assuming that SIFT points located on moving pixels belong to the object of interest with a high probability.

A first reliability measure based on these selection criteria will be useful during the data association step.

2.2. SIFT feature tracking by particle filters

All SIFT features are tracked over time using a particle filtering method. As a reminder about particle filters, let x_t denote the state of the system at the current time *t*, and $y^t = (y_1, \dots, y_t)$ the observations up to time *t*. For tracking, the distribution of interest is the filtering distribution $p(x_t|y^t)$. In Bayesian sequential estimation this distribution can be computed using the two step recursion:

$$\underline{\text{predict}} \quad p(x_t|y^{t-1}) = \int D(x_t|x_{t-1})\, p(dx_{t-1}|y^{t-1}) \qquad (1)$$

$$\underline{\text{update}} \quad p(x_t|y^t) = \frac{L(y_t|x_t)p(x_t|y^{t-1})}{\int (x_t|s_t)\, p(ds_t|y^{t-1})} \qquad (2)$$

where the prediction distribution follows from marginalization and the new filtering distribution is a direct consequence of Bayes' rule. The recursion requires the specification of a dynamic model describing the state evolution $D(x_t|x_{t-1})$, and a model giving the likelihood of any state in the light of the current observation $L(y_t|x_t)$. The recursion is initialized with some initial distribution $p(x_0)$.

In our approach, the state of a SIFT feature $\boldsymbol{x} = \{x, y, u, v, \boldsymbol{h}, n\}$ consists of the SIFT feature position *(x, y)*, the velocity component *(u, v)*, the SIFT descriptor $\boldsymbol{h}$ associated to the SIFT point, and finally *n*, the measurement error estimation following a normalized distribution. In particle filtering, each hypothesis about the new state is represented by a particle which has its own state with the same structure as that of the SIFT feature. Each SIFT feature is tracked using a constant number of particles.

The prediction step consists in applying the dynamic model to all the particles of the tracked SIFT feature to compute the new estimated location of each one:

$$(x, y)_t = (x, y)_{t-1} + (u, v)_{t-1}.\Delta_t + n_{(x,y)} \qquad (3)$$

$$(u, v)_t = (u, v)_{t-1} + n_{(u,v)} \qquad (4)$$

The update step consists in estimating the new location of the tracked feature using the predicted state of all particles. This step is performed in three sub-steps (particle weighting, particle sampling and new state estimation) described below.

2.2.1 Weighting of particles

Each particle is weighted using two different weights (eq 5): the first and most important is the similarity score between the particle descriptor and the tracked feature descriptor. The second weighting criterion aims to minimize the importance of particles on the background using the "motion state" of pixels at the same positions as the particles.

$$Wp = c\left[\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d(H_f,H_p)^2}{2\sigma^2}}\right] \qquad (5)$$

$d(H_f, H_p)$ denotes the similarity between the tracked feature descriptor and the current particle descriptor. We use a Euclidean distance after having tested some other distances without getting significant improvements. σ denotes a standard deviation computed on tracked feature similarity variations up to time *t-1*. $c \in \{c_0, 1\}$, with $c_0 \in]0, 1[$, denotes the coefficient of confidence of the particle according to its belonging to the moving region. At a given time *t* and for a given object, *c* can take two values: 1 if the pixel corresponding to the particle is a motion pixel and a smaller value, c_0 otherwise. c_0 depends on the quality of the detection

of the object measured by the density of its motion pixels. For a low density, c_0 will be close to 1. A high density is obtained in a good or noisy detection case. Here c_0 will be smaller. In fact when the motion detection performs well, non-motion pixels are probably background ones. In the high noise case the object of interest is probably fully detected, so this weight will have low impact in (5). When the image resolution is high and the tracked objects enough large in the image, *c* would serve no purpose; the robustness of the SIFT descriptor is sufficiently discriminative. However, in practice, video-surveillance images have medium resolution quality and are relatively noisy. They monitor large areas making objects smaller. For this reason, our second weighting technique increases accuracy, as shown in sec 3.1.2.

2.2.2 Sampling particles

After weighting, all particles are sampled using a "Sampling Importance Re-sampling" (SIR) method [26, 27] to keep the most important, drop the less important and replace them by new particles generated from the kept ones. The sampling step allows the tracker to keep the more reliable particles and the re-sampling step avoids information degeneration. Each feature keeps a constant number of particles over time, which makes the processing time easier to control. Finally, all particles are re-weighted with the same normalized weight.

2.2.3 New state estimation

The estimation of the new location of the tracked feature is obtained as the barycentre of all its particles. The descriptor of the tracked feature is computed around the new location.

A variation measure between the previous descriptor and the new one is computed. This variation measure is used for the feature variation learning in order to decide if a new state is acceptable. If the variation is too important the SIFT feature is dropped and replaced by a new detected one.

2.3. Data association

At this point all of the tracked features have been updated. The next step consists in linking previously tracked objects *to(t-1)*, with new detected objects *do(t)*, while dealing with complex situations like partial or full occlusions. From a given frame to the next one, only four cases can occur:

In the first case a unique *do(t)* corresponds to only one *to(t-1)*. Here the system updates the *to(t-1)* by linking it directly to *do(t)*.

In the second case a unique *do(t)* corresponds to a set of *Q* $to_k(t\text{-}1)$ where $k \in [1, Q]$. This situation occurs when the detection at time *t* did not correctly split detected moving objects, typically during partial occlusions or high object proximity. Here the system tries to split the bounding box of *do(t)* into *Q* smaller bounding boxes. This split is performed by estimating the best bounding boxes according to the spatial distribution of the SIFT points before the merge. This distribution is given by the ratios between feature locations and the borders of the bounding before detection merging.

In the third case a unique *to(t-1)* corresponds to a set of *R* $do_l(t)$, where $l \in [1, R]$, like in the dispersion of a group of persons, the end of short occlusion or a person leaving a car. Here two situations can be distinguished: $to_i(t\text{-}1)$ can be the result of a previous merge of tracked objects at a time *t-p* like described in the previous paragraph. In this case, the tracking is resumed using the occlusion management approach (sec 2.4). Otherwise, if $to_i(t\text{-}1)$ has always been tracked as a group of objects since its appearance in the scene, new $to_l(t)$ are initialized by each $do_l(t)$ after the split.

In the last case no *do(t)* corresponds to a *to(t-1)*. It occurs in full occlusion situations or when the *to(t-1)* leaves the scene. According to criteria like scene exit proximity or a detected intersection between several *to(t-1)*, the system considers this object as lost or as occluded. If the object is lost, its tracking is definitively stopped. Otherwise, the object is kept for tracking resumption if it re-appears after an occlusion.

The first step of our data association method consists in detecting in which case each *to(t-1)* is it at time *t*. To do this, an M×N link score matrix, denoted **S**, is constructed. M is the number of *to(t-1)* and N the number of *do(t)*. Each element $s(to_i(t-1), do_j(t))$ of **S** is calculated as the weighted proportion of SIFT features from the i^{th} *to(t-1)* that geometrically belongs to the j^{th} *do(t)*. The contribution of each SIFT features in the link score value is directly proportional to its reliability. This reliability is given by the learned similarity variation of the tracked feature up to time *t*, and by the motion state of the pixel at the same location:

$$s(to_i(t-1), do_j(t)) = \frac{1}{P} \sum_{k=1}^{P} w_k(i,j) \quad (6)$$

where $w_k(i,j) \in [0, 1]$ is the reliability score of the k^{th} feature of $to_i(t\text{-}1)$ that is geometrically contained by $do_j(t)$. $to_i(t\text{-}1)$ has *P* SIFT features.

Putting these link score values in a matrix form makes the decision process easier and faster. We use the Hungarian algorithm [28] to select the best links.

Note that after this data association step, SIFT points outside of their objects (moved onto the background or onto other objects during their proximity or partial occlusion) are dropped and replaced by new detected SIFT features. Sub-regions which are common to multiple objects in the case of partial occlusions are not used for the detection. On the other hand, the system keeps a uniform spatial repartition of the SIFT features by filtering out too close features. The system keeps the most reliable feature and replaces others by new detected ones in sub-regions of the object with fewer features.

2.4. Occlusion management

After link creation, some *do(t)* may not be linked with any *to(t-1)*. They can be new objects appearing for the first time in the scene or previously occluded objects which re-appear.

Before initializing new *to(t)* with unlinked *do(t)*, an attempt to match these unlinked *do(t)* to tracked objects in occlusion state is made using the following criteria:

First, a matching between SIFT features used for object tracking before occlusion and new detected ones on the candidate object. In the case of an object which did not

change orientation during occlusion (straight move for example), this matching of SIFT features performs well.

The second criterion is based on the dominant color descriptor. During tracking, the *k* dominant colors[29] of the object are extracted with their proportions and used to weight a matching hypothesis with a candidate object after occlusion.

Finally, we use two "world" coherency criteria, based on the camera calibration information. During tracking, 3D height, 3D width and real speed of the tracked object (computed using camera calibration matrices) are learned in two Gaussian models. For each candidate for resume-after-occlusion, its 3D dimensions and position must fit into the learned Gaussian models.

Note that we keep track of fully occluded objects to potentially resume their tracking only for a limited time. Long period increases the number of combinations and the risk of errors.

2.5. Real object of interest validation

New *to(t)* are initialized for all free *do(t)* after links creation and occlusion management,. A set of SIFT features are detected and assigned to these objects (See sec. 2.1).

New *to(t)* stay in intermediate state before their full validation. Some of the *do(t)* can be noise, such as illumination changes reflected on the floor or on some static scene objects, or foliage movements. For this reason, each new *to(t)* is tracked normally, but controlled during a given number of frames before considering it as a real object of interest. Our system uses the persistence, the trajectory, and the 3D speed of each new *to(t)* during 10 frames at least as criteria to validate it as an object of interest. In the case of noise, the new *to(t)* can disappear after a few frames. It can have incoherent or oscillatory motion or a high speed which cannot match the possible speed of any object of interest.

3 Experimental results

We evaluated the tracking algorithm on 121 sequences from four datasets: CAVIAR[30], ETISEO[31], PETS2001[32] and VS-PETS2003[33]. We have selected these sequences according to the availability of their ground truth data. They contain different levels of complexity with challenging situations, such as football match in VS-PETS2003 dataset.

In order to compare our tracker with another one on the ETISEO dataset, providing the largest variety of situations and contexts, we used the tracking evaluation metrics defined in the ETISEO benchmark project (A.T.Nghiem et al., 2007).

The "tracking time" metric $M1$ measures the percentage of time during which a reference object (ground truth data) is tracked. The "object ID persistence" metric $M2$ computes throughout time how many tracked objects are associated with one reference object. The third metric $M3$ "object ID confusion" computes the number of reference object IDs per tracked object. These metrics must be used together to obtain a complete tracker evaluation. Like in [34], we also use a mean metric M taking the average value of these three tracking metrics. All of the four metric values are defined in the interval [0, 1]. The higher the metric value, the better the tracking algorithm performance.

Our evaluation is divided into two parts. First, we have evaluated our tracker with different configurations and parameters in order to highlight our contributions. The second part shows a comparison with existing evaluation on ETISEO dataset with the same metrics on common sequences.

3.1. Evaluation with different configurations

3.1.1 Detection and selection of SIFT points

In this part, we tried three configurations to evaluate our SIFT point detection and selection method. First, we applied an implementation of SIFT algorithm with the optimum parameters as defined in [12] on the whole objects of interest. In the second configuration, we divided objects into sub-regions and we used the SIFT algorithm with more permissive parameters and selected the needed number of points per sub-region according to their detection order. The third configuration is the one we used for our approach (Sec 2.1) with the following parameter values: 0.005 for contrast threshold and 7.5 for curvature threshold.

		PETS 2001	VS-PETS 2003
Configuration 1	$\overline{M}$	0.43	0.18
Configuration 2	$\overline{M}$	0.65	0.41
Configuration 3	$\overline{M}$	0.69	0.48

Table 1: evaluation of different SIFT feature detection and selection methods on PETS2001 and VS-PETS2003 datasets

We used 2 datasets: in PETS2001, some persons are partially occluded by passing vehicles. In VS-PETS2003, the football match provides a lot of partial occlusions between players.

Table 1 shows that our SIFT detection and selection improves results in comparison to the other tested configurations. The first reason is the number of detected points. We observed that for configuration 1, the SIFT algorithm provides very low number of SIFT points, due to the small size of objects in the images, and the image resolution. This makes the data association less precise. Our method (configuration 3) provides more points thus improving the robustness of data association process.

The second reason is the localization of the detected points. For configuration 1, most detected points are concentrated on the feet of tracked persons. The concentration of tracked SIFT features in one region of the tracked object makes the tracking fail if this region is occluded. The improvement of sub-regions division is demonstrated by the results of configuration 2 and 3. The spatial distribution of features, even if some are less reliable, ensure existence of some points on visible parts of objects in partial occlusions.

The last reason is the selection of SIFT features according to their reliability. In configuration 2, we take the *n* first detected SIFT points; *n* being the number of points per sub-region. This increases the risk of taking less reliable points instead of

more reliable ones. In our method (configuration 3), we select the most reliable points as described in section 2.1.

The improvements of our feature detection and selection approach are illustrated in Figure 2.

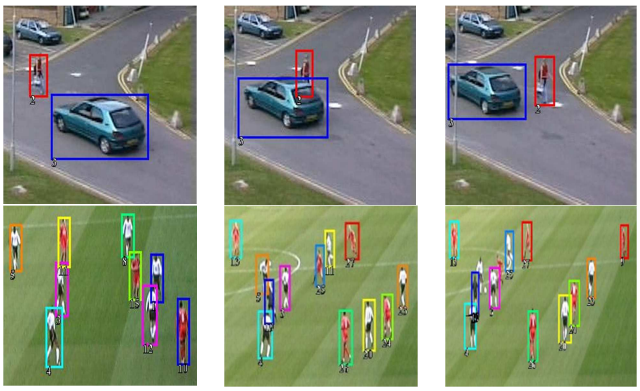

Figure 2: Partial and full occlusion management

3.1.2 Particle weighting using motion state

To highlight the contribution of weighting particles using the motion state given by *c* in eq. (5), we have taken a subsequence of 50 frames from the PORTUGAL-FV sequences of CAVIAR dataset. It contains a person crossing the scene in straight line. We apply the SIFT algorithm on this person and select one SIFT feature on its head. After that we manually annotate the approximate location of this SIFT point on the 49 remaining frames. Finally, we track this point along the subsequence using the described particle filter (Sec 2.2) with and without using the weighting method by motion state (Sec 2.2.1).

- a - **- b -** **- c -** **- d -**

Figure 3: different qualities of motion detection. (a) Original image. (b) Low detection. (c) Medium detection. (d) High detection.

To make this test more relevant, we change some parameters in the motion detection algorithm so that it provides three different qualities of detections (see Figure 3).

We have observed that when we do not use the motion state of pixels for particle weighting, the SIFT point detaches itself and stays on the background after the 32th frame. This is due to the successive updates of the SIFT descriptor during its tracking. Starting from the 24 frame, the SIFT point is located too close to the contour of the person's head, so the computing window of the SIFT descriptor takes more information from the background. The SIFT point continues to diverge, attracted by the background, until the 32th frame where it will indefinitely stick to it. On the other hand when we use the motion state for particles weighting, the SIFT point stays all the 50 frames on the head of the person.

3.1.3 Data association and occlusion management

We assume that the acceptance of less reliable SIFT features to ensure spatial distribution can decrease the reliability of object localization. At the same time, our use of motion state of pixels to weight particles decreases slightly the final weight of each particle, making the SIFT feature move a little bit more around its real position (see Table 2).

	Without motion state weighting	With motion state weighting
Low detection	5.59	6.12
Medium detection	5.59	6.72
High detection	5.59	6.04

Table 2: divergence of SIFT point until frame 32: the mean of 2D distance between tracked SIFT point and annotated position

Our data association approach compensates the unreliability of SIFT features in this case. Using reliability measure of SIFT features as a weight in link scores allows the algorithm to select the right links, and drop unreliable SIFT features.

Table 3 validates this method and our tracking framework.

	M1	M2	M3	$\overline{M}$
CAVIAR	0.78	0.82	0.91	**0.84**
ETISEO	0.7	0.91	0.92	**0.84**
PETS2001	0.84	0.90	0.94	**0.89**
VS-PETS2003	0.47	0.79	0.84	**0.70**

Table 3: global evaluation results on the selected 121 sequences.

3.2. Comparison with state of the art results

We compared the results of our approach on the ETISEO dataset with the results of [34] who obtained better results than those of ETISEO. The comparison is provided in Table 4. We obtained better results on the same sequences with the same metrics for most of them.

		ETI-VS1-BE-18-C4	ETI-VS1-BE-16-C4	ETI-VS1-MO-7-C1
Proposed tracked	M1	**0.68**	**0.54**	**0.90**
	M2	**1**	**1**	0.89
	M3	**1**	**1**	**1**
	$\overline{M}$	**0.89**	**0.85**	**0.93**
T_{CHAU} [34]	M1	0.64	0.36	0.87
	M2	**1**	**1**	0.92
	M3	**1**	**1**	**1**
	$\overline{M}$	0.88	0.79	**0.93**

Table 4: Comparison of proposed tracker performances with the one proposed by CHAU et al. [34] on three ETISEO sequences

Note that the average running time of our code is 4–8 fps for ETISEO, PETS2001 and VS-PETS2003 datasets, and 12-32 fps for CAVIAR dataset, with Intel(R) Xeon(R) E5530 2.40GHz, depending on the image size, number and 2D size of detections in each sequence.

4 Conclusion

The main idea presented in this paper is that the correct use of local features as long as they are wisely selected and reliably tracked and used in the data association technique by correct weighting, can solve most of object tracking issues.

Many works aim at solving the problems given by the tracking process (such as occlusion), but a robust tracker still does not exist which can deal correctly with all possible situations.

The proposed approach has been tested and validated on 121 real video sequences from four different datasets. The experimentation results show that the proposed tracker provides good results in many scenes although each tested scene has its proper complexity. Our tracker also gets better performances than other recent approaches [34].

Our algorithm processes in real-time. However, some drawbacks still exist in this approach: the use of motion detection as a unique input for our tracker slows down the tracking time and segments the trajectories of objects remaining static for a long time. Adding object detectors on static images (people detector, car detector) can limit this kind of problems. For occlusion management, our criteria for candidate validation provide good results but can be improved by more reliable descriptors (e.g. color covariance). All these improvements are in track for our future works.

References

[1] C.H Kuo and C. Huang and R. Nevatia. "Multi-target tracking by online learned discriminative appearance models". In the IEEE CVPR, San Francisco, CA, USA, June 2010.

[2] B. Leibe, K. Schindler, and L. Van Gool. "Coupled detection and trajectory estimation for multi-object tracking". In ICCV,2007.

[3] A. P. Leung and S. Gong. "Mean-shift tracking with random sampling". In British Machine Vision Conference (BMVC), Edinburgh (UK), September 4-7 2006.

[4] L. Snidaro and I. Visentini and G.L. Foresti. "Dynamic models for people detection and tracking". In the 8th IEEE AVSS, pp. 29–35, September 2008.

[5] A. Yilmaz and O. Javed and M. Shah. "Object tracking: A survey". Volume 38 Issue 4, December 2006

[6] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L.V. Gool. "Robust Tracking-by-Detection using a Detector Confidence Particle Filter", In the 12th IEEE ICCV, Kyoto, Japan, Oct-2009.

[7] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". CVPR, 2005.

[8] B. Leibe, A. Leonardis, and B. Schiele. "Robust object detection with interleaved categorization and segmentation". IJCV, 77(1-3):259–289, 2008.

[9] F. Moutarde, B. Stanciulescu, A. Breheret, "Real time visual detection of vehicles and pedestrians with new efficient adaboost features," IEEE International Conference on Intelligent Robots Systems (IROS 2008), September 2008.

[10] B. Wu and R. Nevatia. "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors". IJCV, 75(2):247–266, 2007

[11] N.T. Siebel and S.Maybank, "Fusion of Multiple Tracking Algorithms for Robust People Tracking". In Proc. ECCV, 373—387, 2002

[12] D. Lowe. "Distinctive image features from scale-invariant keypoints". IJCV, 60(2):91–110, 2004.

[13] Y. Ke and R. Sukthankar. "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors". Proc.CVPR 2:506–513. 2004

[14] K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors". IEEE Transactions on Pattern Analysis and Machine Intelligence, 27:1615–1630, 2005.

[15] E. Tola, V. Lepetit, and P. Fua. "A Fast Local Descriptor for Dense Matching". CVPR 2008

[16] H. Bay, T. Tuytelaars, and L.V. Gool. "SURF: Speeded up robust features." ECCV, 2006.

[17] M. Calonder, V. Lepetit, C. Strecha, P. Fua, "BRIEF: Binary Robust Independent Elementary Features", 11th ECCV, Heraklion, Crete, Springer, September 2010.

[18] N. Gordon, D. Salmond, and A. Smith. "Novel approach to non-linear/non-Gaussian Bayesian state estimation". IEE Proceedings-F, 140(2):107–113, 1993.

[19] M. Isard and A. Blake. "Condensation—conditional density propagation for visual tracking". IJCV, 29(1):5–28, 1998.

[20] G. Kitagawa. "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models". Journal of Computational and Graphical Statistics, 5(1):1–25, 1996.

[21] K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. Lowe. "A boosted particle filter: Multitarget detection and tracking". In ECCV, 2004.

[22] J. Vermaak, A. Doucet, and P. Perez. "Maintaining multimodality through mixture tracking". In ICCV, 2003.

[23] A. Almeida, J. Almeida, and R. Araujo. "Real-time tracking of multiple moving objects using particle filters and probabilistic data association". Automatika, vol. 46, no. 1-2, pp 39-48, 2005.

[24] Y. Rui and Y. Chen, "Better Proposal Distributions: Object Tracking Using Unscented Particle Filter". In the IEEE CVPR. Kauai, HI, USA. 2001.

[25] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An Adaptive Color-Based Particle Filter", Image and Vision Computing, Vol. 21, Issue 1, pp. 99-110, Jan 2003

[26] Donald B. Rubin. "The calculation of posterior distributions by data augmentation". Jornal of the American Statistical Association, 82, 1987.

[27] A. F. M. Smith and A. E. Gelfand. "Bayesian statistics without tears: A sampling–resampling perspective". The American Statistician, 46(2):84–88, May 1992.

[28] H. W. Kuhn, "The Hungarian method for the assignment problem," Naval Res. Logistics Quart. 2, 83–97 _1955_

[29] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 6, pp. 703–715, Jun 2001.

[30] CAVIAR: Context Aware Vision using Image-based Active Recognition, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[31] ETISEO: Video understanding Evaluation, http://www-sop.inria.fr/orion/ETISEO/.

[32] PETS'2001 The Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Kauai, Hawaii. December 2001. http://visualsurveillance.org/PETS2001/

[33] VS-PETS'2003 The First Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (Nice, October 2003). http://vspets.visualsurveillance.org/

[34] D. P. Chau, F. Bremond, M. Thonnat and E. Corvee, "Robust mobile object tracking based on multiple feature similarity and trajectory filtering", in The International Conference on Computer Vision Theory and Applications (VISAPP), Algarve, Portugal, March 5-7, 2011